# Data Structures and Algorithms Project Evaluation Sheet

**Name**: Shahin Mirakhan **SRN**: 01FE23BCS411 **Roll no** :265

**Implementation Analysis:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Algorithm/Data Structure** | **Used? (Yes/No)** | **How and where?** | **Space Efficiency** | **Time Efficiency** |
| Arrays | Yes | In coaching mechanism | Train Route | Access: O(1) |
| Structures | No | - | - | - |
| List | No | - | - | - |
| Stack | No | - | - | - |
| Queue | Yes | For real time updates of trains | O(V) | Priority queue: O(V^2) |
| Binary Tree | No | - | - | - |
| Binary Search Tree | Yes | For user preference and requirement | Average:O(numRoutes) | Search: Average O(log n) |
| AVL Tree | Yes | For user preference and requirement | Average:O(numRoutes) | Search: Average O(log n) |
| 2-3 Tree | No | - | - | - |
| Red-Black Tree | Yes | Load balancing | O(n) | O(n) |
| Trie | No | - | - | - |
| Heap | Yes | Priority queue in real time updates for latest upadte | O(V) | Priority queue: O(V^2) |
| Lookup Table | No | - | - | - |
| Sparse Table | No | - | - | - |
| Fenwick Tree | No | - | - | - |
| Segment Tree | No | - | - | - |
| Skip List | No | - | - | - |
| Union-Find | No | . | - | - |
| Hashing | Yes | For station information storage | O(1) | O(n + m) |
| DFS | No | . | - | - |
| BFS | No | . | - | - |
| Bubble Sort | No | - | - | - |
| Selection Sort | No | - | - | - |
| Insertion Sort | No | - | - | - |
| Quick Sort | No |  | - | - |
| Merge Sort | No | - | - | - |
| Brute Force String Search | No | - | - | - |
| Rabin Karp | No |  | - | - |
| Boyer-Moore | No | - | - | - |
| Knuth-Morris-Pratt | No | - | - | - |
| Heap Sort | No | - | - | - |
| Kruskal | Yes | To find optimal trasnfer point | O(1) | Avg:O(E \* log(V)) |
| Prim | Yes | Minimum spanning tree to find min cost | O(1) | Avg:O(E \* log(V)) |
| Dijkstra | Yes | To find shortest path in route planning | O(V^2) | O((V + E) \* log(V)) |
| Floyd | No | - | - | - |
| Warshall | No | - | - | - |
| Bellman-Ford | No | - | - | - |
| Any Other | Yes | String for route identify | O(numStations) | O(numStations) |

# Other Analysis:

Number of Lines of Code Written: 1406 Number of Functions: 43

Size of Largest File: 10 Kb Number of files created: 10

**Design Techniques and Principles used**:

- **Modular Design**: The project is divided into distinct modules as per separate functionalities.

- **Data Abstraction**: Using structures to represent complex entities like Stations, Routes, .

- **Algorithm Optimization**: Implementing efficient algorithms like Kruskal’s, Dijkstra's, and Prim’s for specific tasks.

- **Encapsulation**: Functions are used to encapsulate specific functionalities.

|  |  |  |
| --- | --- | --- |
| **Si. No.** | **Criteria** | **Marks** |
| 1 | Team Evaluation (already completed) | 10 |
| 2 | Design and functionality Identification | 5 |
| 3 | Algorithms, DS and Principles used | 15 |
| 4 | Efficiency Analysis | 10 |
| 5 | Coding Standards and Organization | 5 |
| 6 | Report | 5 |
|  | **Total** | 50 |